



## UNITED STATES PATENT AND TRADEMARK OFFICE

### Facsimile Transmission

To:           Name:                   Boris Matvenko (reg. 48165)  
              Company:  
              Fax Number:       2129833115  
              Voice Phone:

From:       Name:  
              Voice Phone:

37 C.F.R. 1.6 sets forth the types of correspondence that can be communicated to the Patent and Trademark Office via facsimile transmissions. Applicants are advised to use the certificate of facsimile transmission procedures when submitting a reply to a non-final or final Office action by facsimile (37 CFR 1.8(a)).

#### Fax Notes:

---

Attached are proposed amendments to try to place this case into condition for allowance. Please let me know by Friday, June 5, 2009 at 12pm (eastern time zone) whether the amendments are accepted, so that an examiner's amendment may be done.

---

Date and time of transmission: Wednesday, June 03, 2009 10:24:18 AM  
Number of pages including this cover sheet: 06

---

Claims 1-12 (cancelled)

Claim 13 (Currently Amended). A method for storing data, the method comprising:

providing a fileserver having:

a file system ~~[[operative]]~~ configured to store client files;

a policy component configured to store a protection policy associated with a set of files;

a mirror service in communication with the policy component, the mirror service ~~[[operative]]~~ configured to prepare modified and created files in a set of files to be written to a repository as specified in the protection policy associated with the set of files;

a fileserver API coupled to the mirror service and configured to communicate with a repository;

a fileserver file transfer module in communication with the file system and configured to transfer files for the file system to ~~[[and/or]]~~ or from at least one repository; and,

a location updating component configured to maintain a list of repository nodes that contain a replica of each file in the set of files and a list of files in the set of files stored at the destination fileserver;

said fileserver is configured to initiate recovery of files in the set of files on the fileserver, wherein based on the list of files and the list of repository nodes stored at said fileserver, a replica of a file in the list of files is recovered from a repository node in the list of repository nodes;

wherein using a stub file in the set of stub files[[,]] said fileserver is configured to allow access to a full content of a file associated with the stub file by receiving a client request for a specified file in the set of files, replacing the stub file with the full content of the specified file associated with the stub file, and replacing remaining stub files in the set of stub files with respective full contents of remaining files in the set of files while replacing the stub file with the full content of the specified file;

determining a caching level for said fileserver; and  
recursively, determining a utilization of the fileserver;  
comparing the caching level against the utilization; and  
creating a file migration candidate list when the utilization exceeds the caching level;  
staging out one candidate file;  
replacing the candidate file with a stub file; and  
determining whether the utilization of the fileserver still exceeds the caching level, wherein said determining whether the utilization of the fileserver still exceeds the caching level further comprises staging out another candidate file on the candidate list and again determining if the utilization of the fileserver exceeds the caching level.

Claims 14-19 (cancelled)

Claim 20 (Previously Presented). The method according to claim 13, wherein the set of files is the set of files that have been accessed during a specified period; and wherein the recovery

service is further configured to recursively replace the stub files associated with the files that were accessed within the specified period until all stub files associated with the set of files have been replaced.

Claim 21 (new). A system for storing data, the system comprising:

a fileserver having:

fileserver hardware;

a file system configured to store client files;

a policy component configured to store a protection policy associated with a set of files;

a mirror service in communication with the policy component, the mirror service configured to prepare modified and created files in a set of files to be written to a repository as specified in the protection policy associated with the set of files;

a fileserver API coupled to the mirror service and configured to communicate with a repository;

a fileserver file transfer module in communication with the file system and configured to transfer files for the file system to and/or from at least one repository; and,

a location updating component configured to maintain a list of repository nodes that contain a replica of each file in the set of files and a list of files in the set of files stored at the destination fileserver;

said fileserver is configured to initiate recovery of files in the set of files on the fileserver, wherein based on the list of files and the list of repository nodes stored at said

fileserver, a replica of a file in the list of files is recovered from a repository node in the list of repository nodes;

wherein a stub file in the set of stub files is being used said fileserver is configured to allow access to a full content of a file associated with the stub file by receiving a client request for a specified file in the set of files, replacing the stub file with the full content of the specified file associated with the stub file, and replacing remaining stub files in the set of stub files with respective full contents of remaining files in the set of files while replacing the stub file with the full content of the specified file;

a component to determine a caching level for said fileserver; and

a component to recursively, determine a utilization of the fileserver;

a component to compare the caching level against the utilization; and

a component to create a file migration candidate list when the utilization exceeds the caching level;

a component to stage out one candidate file;

a component to replace the candidate file with a stub file; and

a component to determine whether the utilization of the fileserver still exceeds the caching level, wherein said component to determine whether the utilization of the fileserver still exceeds the caching level further comprises to stage out another candidate file on the candidate list and again determining if the utilization of the fileserver exceeds the caching level.

Claim 22 (new). The system according to claim 21, wherein the set of files is the set of files that

has been accessed during a specified period; and wherein the recovery service is further configured to recursively replace the stub files associated with the files that were accessed within the specified period until all stub files associated with the set of files have been replaced.